LAPORAN NETWORK AUTOMATION MENGGUNAKAN DJANGO

Oleh:

Ketut Manik Yogantara	(2115051073)
Gede Dea Braga Anggita	(2115051064)

S1 PENDIDIKAN TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN KEJURUAN

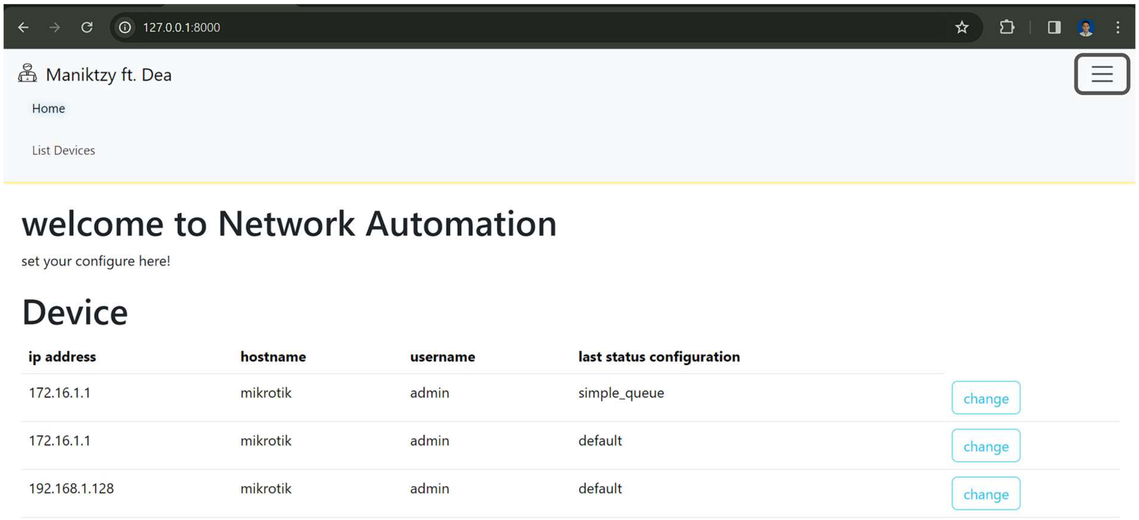
UNIVERSITAR PENDIDIKAN GANESHA

SINGARAJA

2022/2023

Preview Project

1. Tampilan awal web



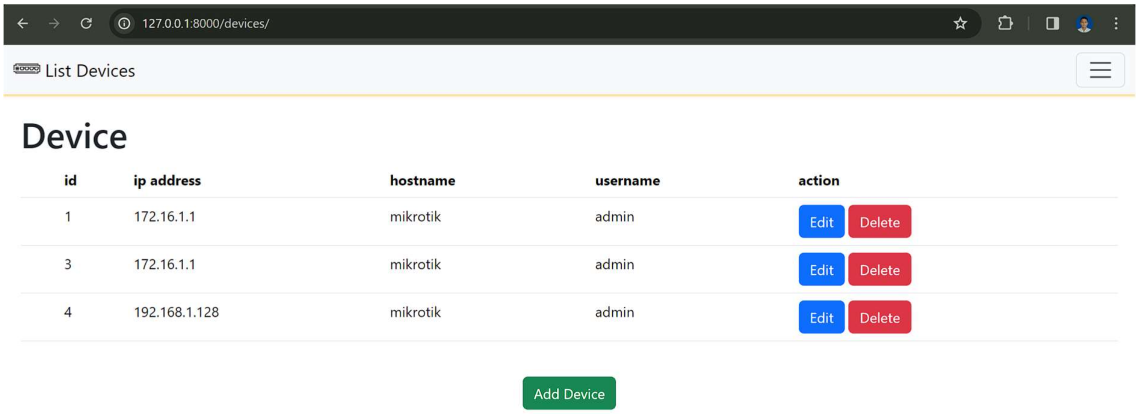
welcome to Network Automation

set your configure here!

Device

ip address	hostname	username	last status configuration	
172.16.1.1	mikrotik	admin	simple_queue	change
172.16.1.1	mikrotik	admin	default	change
192.168.1.128	mikrotik	admin	default	change

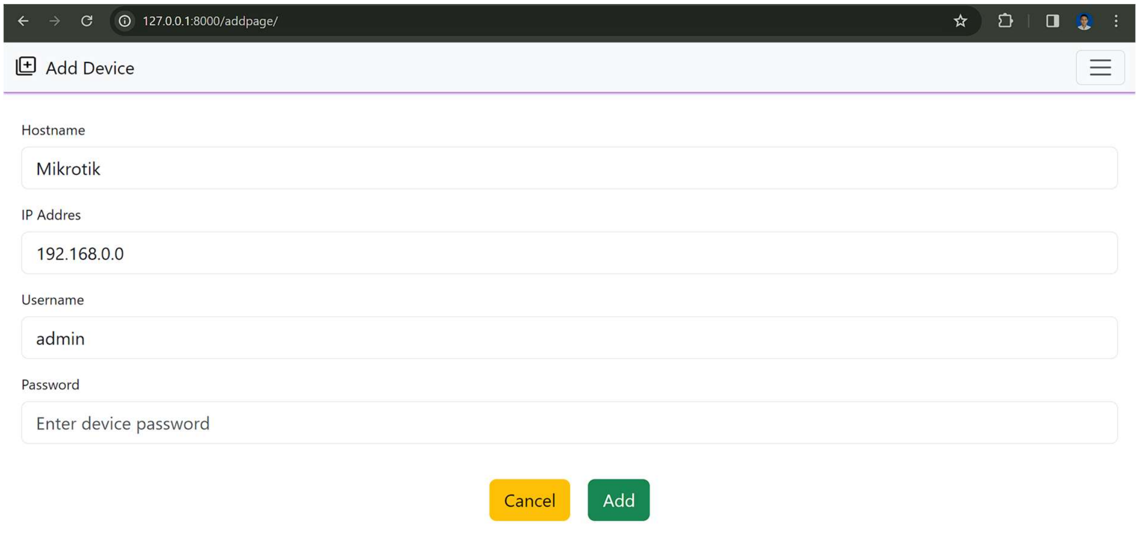
2. Kita bisa menambahkan device baru dengan pergi ke menu List Devices lalu pilih add device



Device

id	ip address	hostname	username	action
1	172.16.1.1	mikrotik	admin	Edit Delete
3	172.16.1.1	mikrotik	admin	Edit Delete
4	192.168.1.128	mikrotik	admin	Edit Delete

3. Isi form berikut sesuai denga ip addres mikrotik anda



Hostname

Mikrotik

IP Addres

192.168.0.0

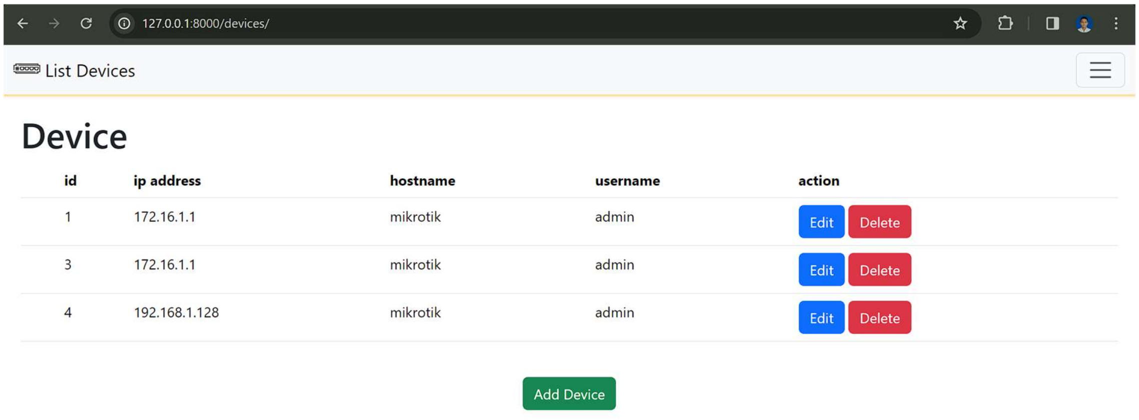
Username

admin

Password

Enter device password

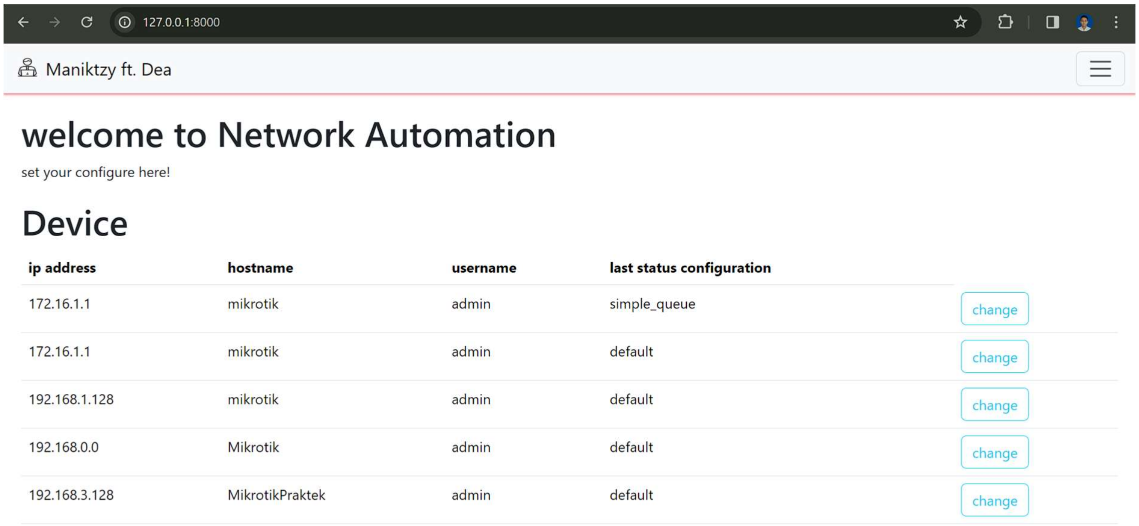
4. Kita dapat mengubah ataupun menghapus perangkat berikut dengan memilih tombol edit atau hapus



Device

id	ip address	hostname	username	action
1	172.16.1.1	mikrotik	admin	Edit Delete
3	172.16.1.1	mikrotik	admin	Edit Delete
4	192.168.1.128	mikrotik	admin	Edit Delete

5. Pilih tombol change pada device yg baru di tambahkan untuk melakukan configurasi



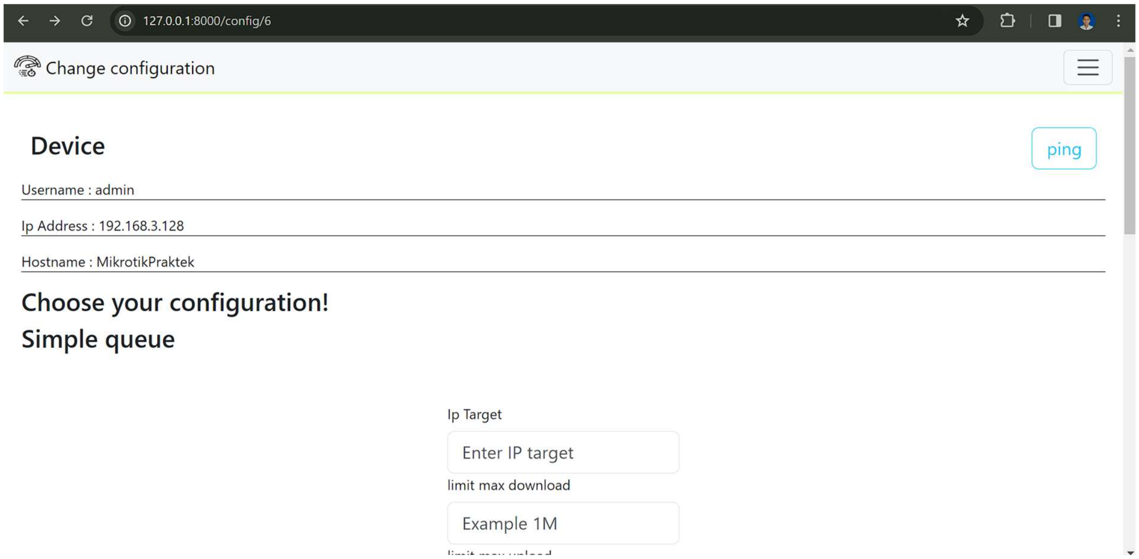
welcome to Network Automation

set your configure here!

Device

ip address	hostname	username	last status configuration	
172.16.1.1	mikrotik	admin	simple_queue	change
172.16.1.1	mikrotik	admin	default	change
192.168.1.128	mikrotik	admin	default	change
192.168.0.0	Mikrotik	admin	default	change
192.168.3.128	MikrotikPraktek	admin	default	change

6. Kita dapat melakukan tes ping untuk device tersebut dengan mengklik tombol ping



Device

Username : admin

Ip Address : 192.168.3.128

Hostname : MikrotikPraktek

Choose your configuration!

Simple queue

Ip Target

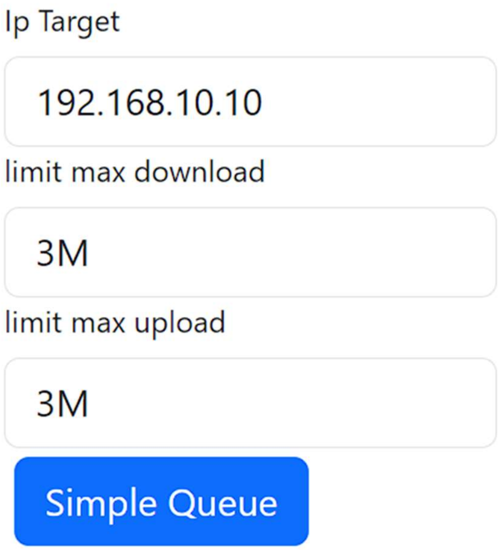
Enter IP target

limit max download

Example 1M

Preview simple queue

1. Geser ke bawah dan anda akan menemukan menu simple queue, isi form sesuai denga ip target dan setting max limit down dan uplodnya seperti berikut, selanjutnya klik simple queue



Ip Target

192.168.10.10

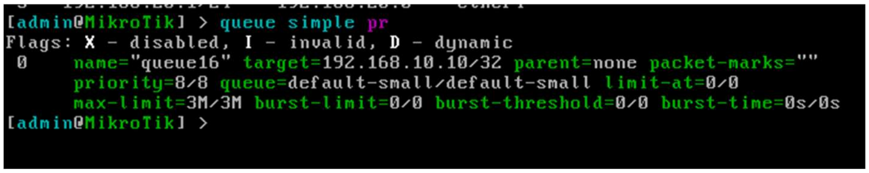
limit max download

3M

limit max upload

3M

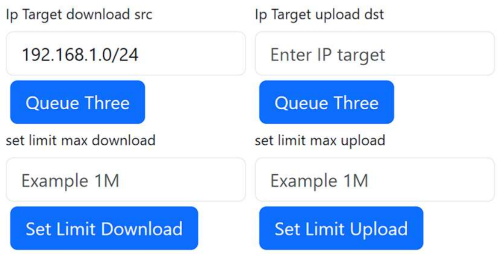
Hasil :



```
[admin@MikroTik] > queue simple pr
Flags: X - disabled, I - invalid, D - dynamic
 0    name="queue16" target=192.168.10.10/32 parent=none packet-marks=""
      priority=8/8 queue=default-small/default-small limit-at=0/0
      max-limit=3M/3M burst-limit=0/0 burst-threshold=0/0 burst-time=0s/0s
[admin@MikroTik] >
```

Preview tree queue

1. Kita dapat memilih ip target untuk max download ataupun max upload, kita juga dapat mensetting nilai dari max upload dan max downloadnya melalui from yang disediakan

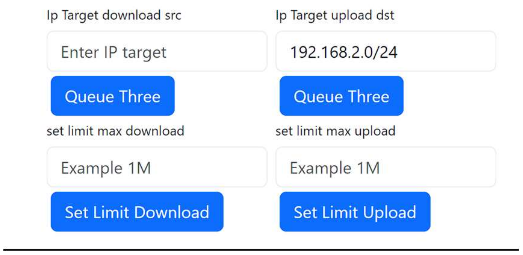


Ip Target download src: 192.168.1.0/24

Ip Target upload dst: Enter IP target

set limit max download: Example 1M

set limit max upload: Example 1M



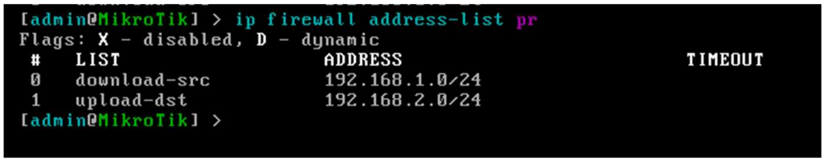
Ip Target download src: Enter IP target

Ip Target upload dst: 192.168.2.0/24

set limit max download: Example 1M

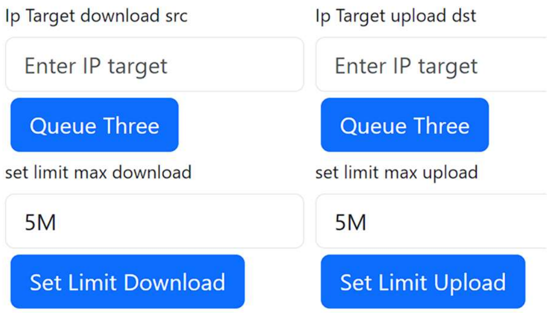
set limit max upload: Example 1M

Hasil :



```
[admin@MikroTik] > ip firewall address-list pr
Flags: X - disabled, D - dynamic
 #   LIST                    ADDRESS                              TIMEOUT
 0   download-src            192.168.1.0/24
 1   upload-dst              192.168.2.0/24
[admin@MikroTik] >
```

2. Merubah setting max limit download dan upload



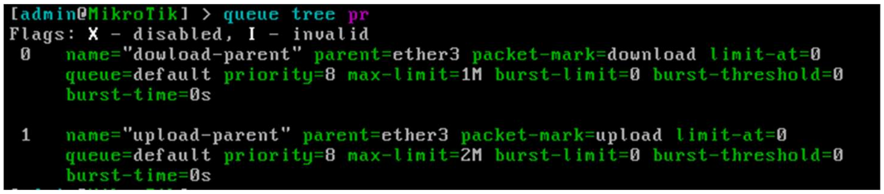
Ip Target download src: Enter IP target

Ip Target upload dst: Enter IP target

set limit max download: 5M

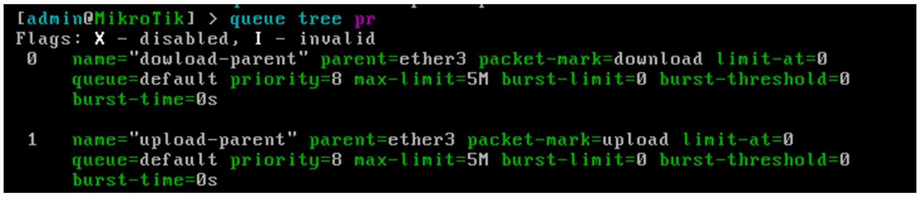
set limit max upload: 5M

Before :



```
[admin@MikroTik] > queue tree pr
Flags: X - disabled, I - invalid
 0    name="dowload-parent" parent=ether3 packet-mark=download limit-at=0
      queue=default priority=8 max-limit=1M burst-limit=0 burst-threshold=0
      burst-time=0s

 1    name="upload-parent" parent=ether3 packet-mark=upload limit-at=0
      queue=default priority=8 max-limit=2M burst-limit=0 burst-threshold=0
      burst-time=0s
```

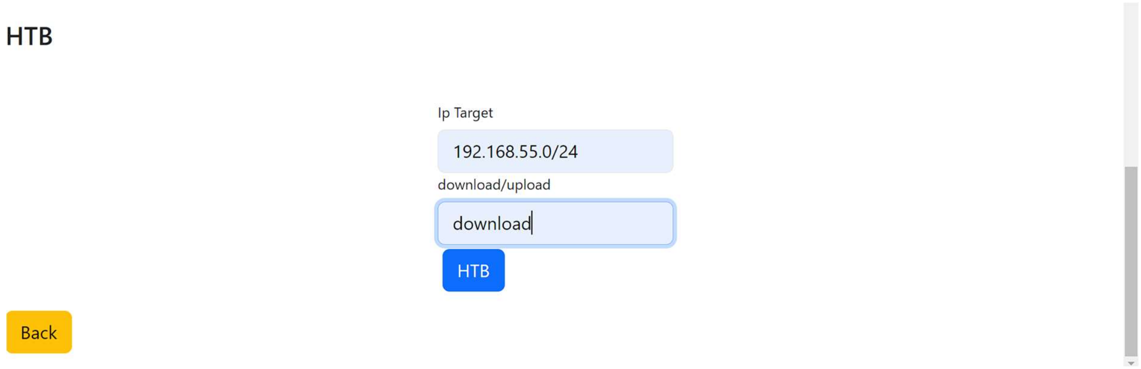
After :



```
[admin@MikroTik] > queue tree pr
Flags: X - disabled, I - invalid
 0    name="dowload-parent" parent=ether3 packet-mark=download limit-at=0
      queue=default priority=8 max-limit=5M burst-limit=0 burst-threshold=0
      burst-time=0s

 1    name="upload-parent" parent=ether3 packet-mark=upload limit-at=0
      queue=default priority=8 max-limit=5M burst-limit=0 burst-threshold=0
      burst-time=0s
```

Preview HTB

1. Masukan form sesuai denga nip target dan pilih download atau upload lalu tekan tombol htb



HTB

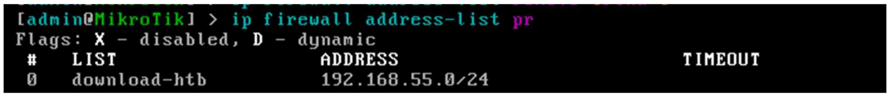
Ip Target

192.168.55.0/24

download/upload

download

2. Hasil

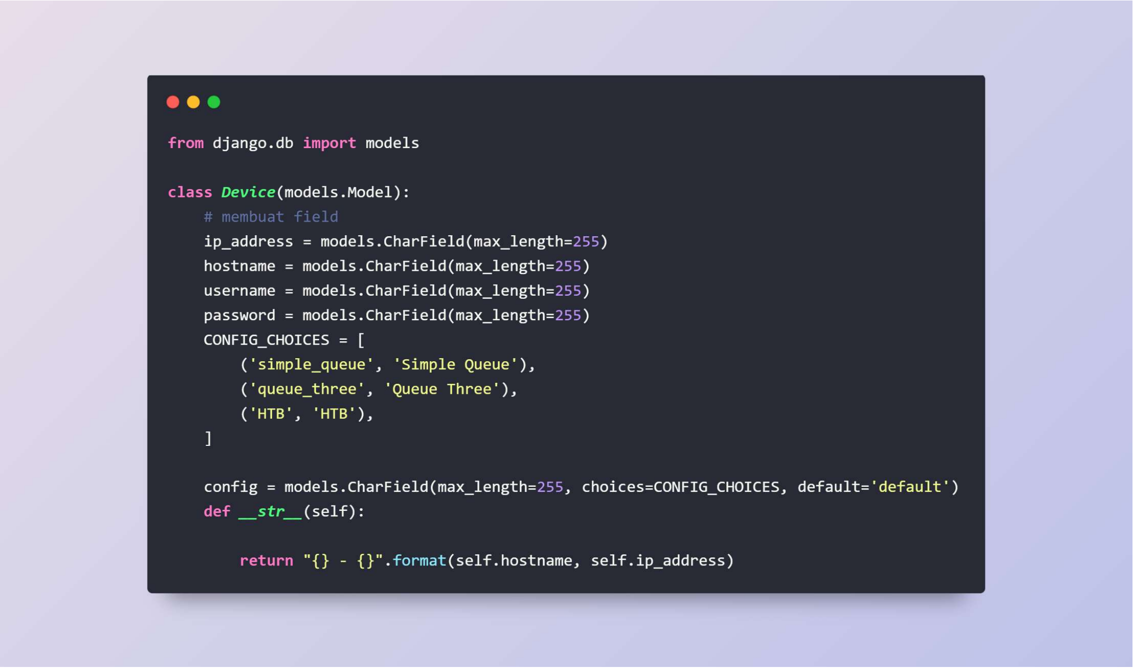


```
[admin@MikroTik] > ip firewall address-list pr
Flags: X - disabled, D - dynamic
 #   LIST                  ADDRESS                                     TIMEOUT
 0   download-htb          192.168.55.0/24
```

Penjelasan Code

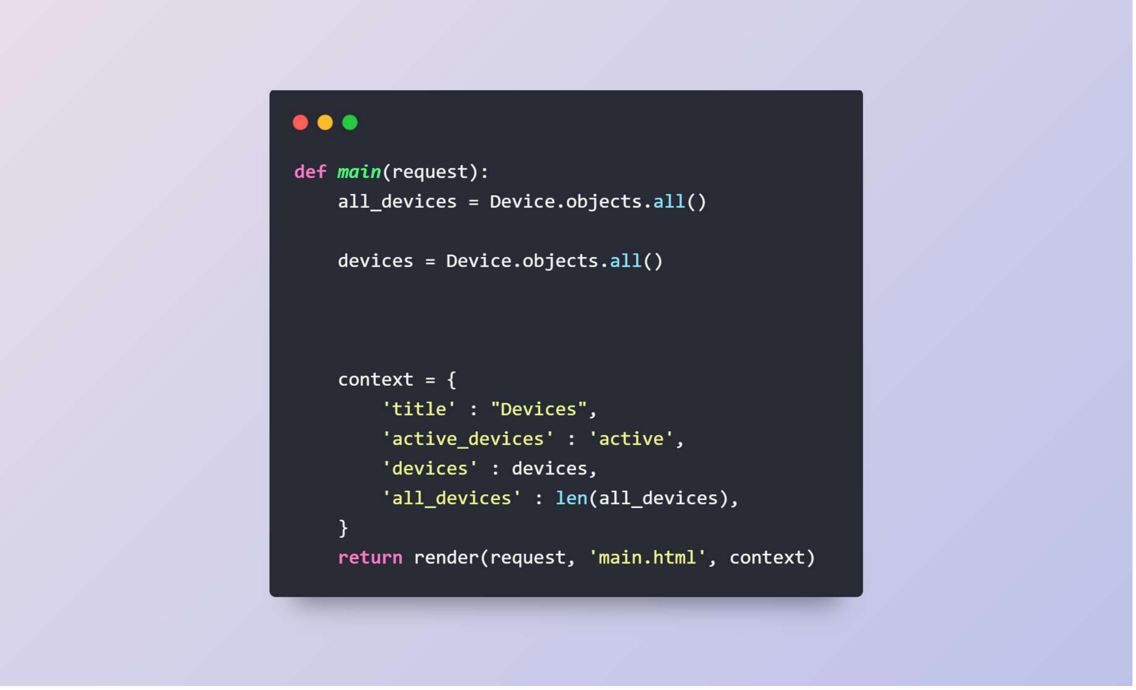
A. CRUD

Database yg dibuat



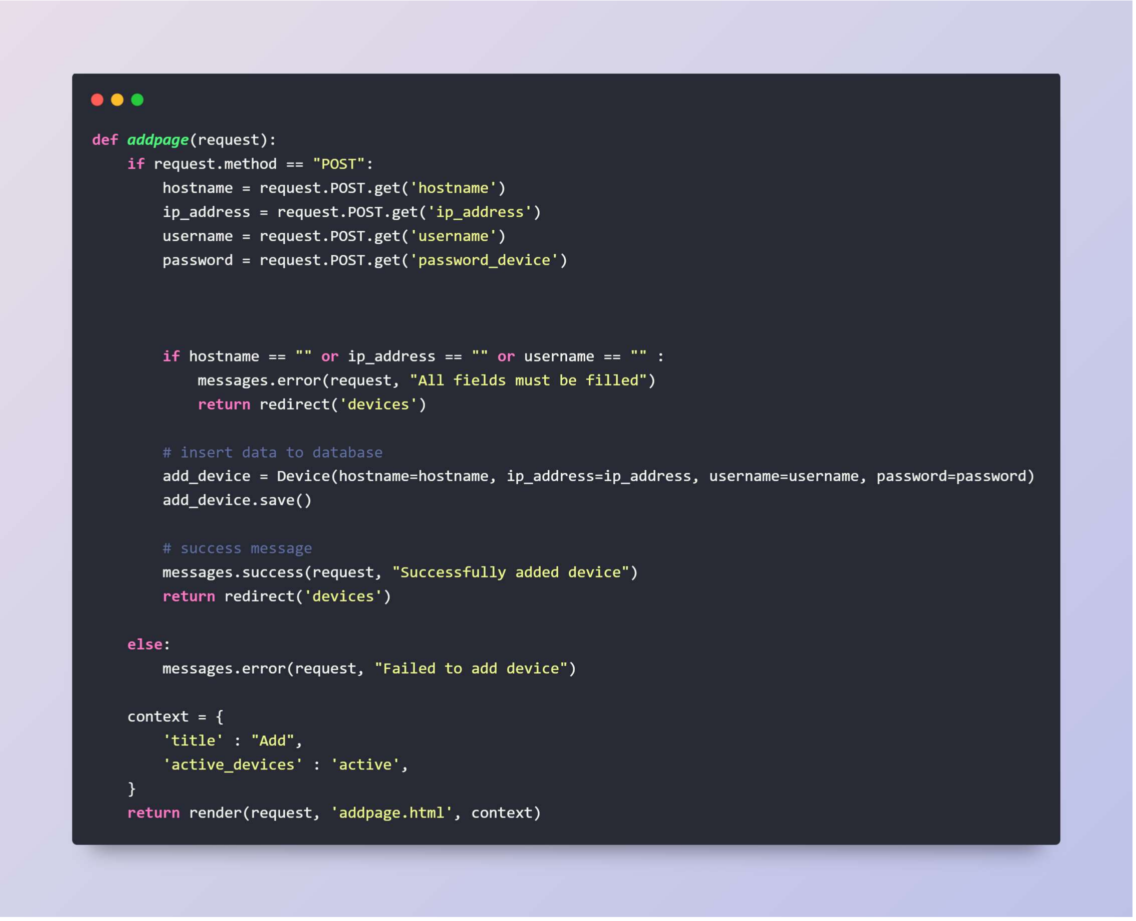
```
from django.db import models

class Device(models.Model):
    # membuat field
    ip_address = models.CharField(max_length=255)
    hostname = models.CharField(max_length=255)
    username = models.CharField(max_length=255)
    password = models.CharField(max_length=255)
    CONFIG_CHOICES = [
        ('simple_queue', 'Simple Queue'),
        ('queue_three', 'Queue Three'),
        ('HTB', 'HTB'),
    ]

    config = models.CharField(max_length=255, choices=CONFIG_CHOICES, default='default')
    def __str__(self):

        return "{} - {}".format(self.hostname, self.ip_address)
```

a) Read



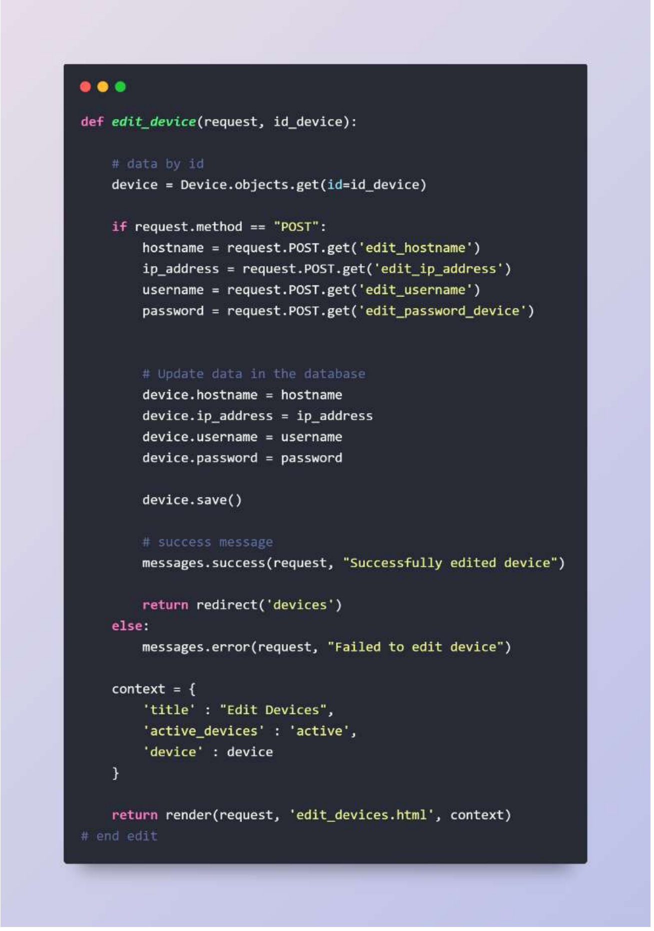
```
def main(request):
    all_devices = Device.objects.all()

    devices = Device.objects.all()



    context = {
        'title' : "Devices",
        'active_devices' : 'active',
        'devices' : devices,
        'all_devices' : len(all_devices),
    }
    return render(request, 'main.html', context)
```

b) Create



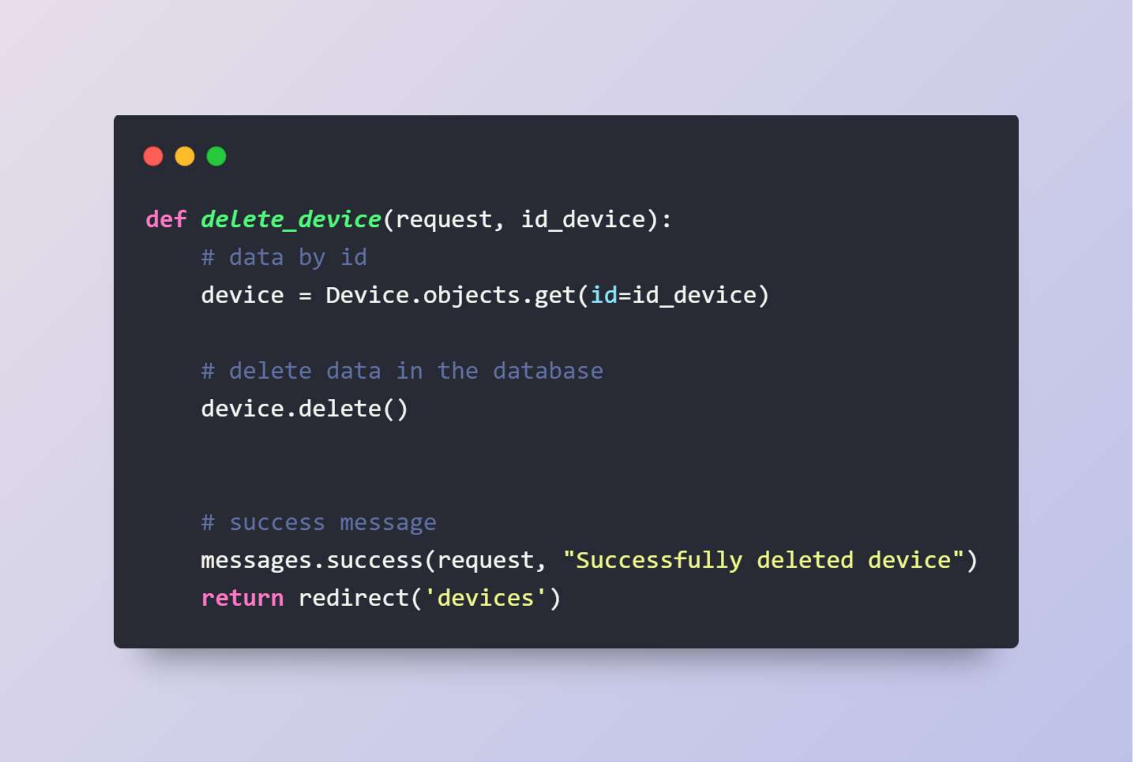
```python
def addpage(request):
    if request.method == "POST":
        hostname = request.POST.get('hostname')
        ip_address = request.POST.get('ip_address')
        username = request.POST.get('username')
        password = request.POST.get('password_device')



        if hostname == "" or ip_address == "" or username == "" :
            messages.error(request, "All fields must be filled")
            return redirect('devices')

        # insert data to database
        add_device = Device(hostname=hostname, ip_address=ip_address, username=username, password=password)
        add_device.save()

        # success message
        messages.success(request, "Successfully added device")
        return redirect('devices')

    else:
        messages.error(request, "Failed to add device")

    context = {
        'title' : "Add",
        'active_devices' : 'active',
    }
    return render(request, 'addpage.html', context)
```

c) Edit



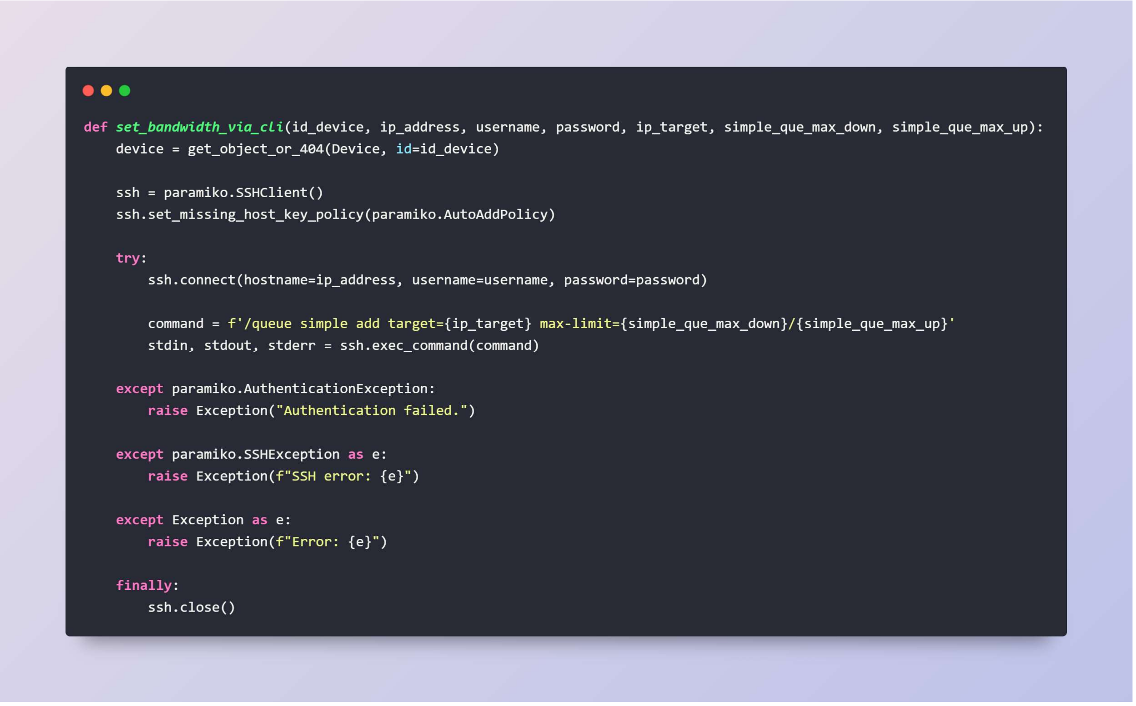
```python
def edit_device(request, id_device):

    # data by id
    device = Device.objects.get(id=id_device)

    if request.method == "POST":
        hostname = request.POST.get('edit_hostname')
        ip_address = request.POST.get('edit_ip_address')
        username = request.POST.get('edit_username')
        password = request.POST.get('edit_password_device')


        # Update data in the database
        device.hostname = hostname
        device.ip_address = ip_address
        device.username = username
        device.password = password

        device.save()

        # success message
        messages.success(request, "Successfully edited device")

        return redirect('devices')
    else:
        messages.error(request, "Failed to edit device")

    context = {
        'title' : "Edit Devices",
        'active_devices' : 'active',
        'device' : device
    }

    return render(request, 'edit_devices.html', context)
# end edit
```

d) Delete



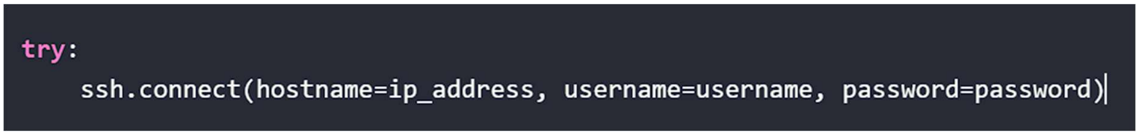
```python
def delete_device(request, id_device):
    # data by id
    device = Device.objects.get(id=id_device)

    # delete data in the database
    device.delete()


    # success message
    messages.success(request, "Successfully deleted device")
    return redirect('devices')
```

B. Simple queue



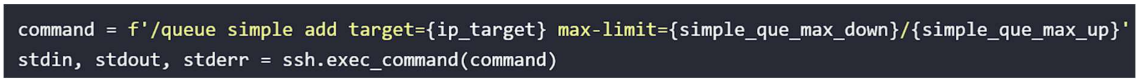
```python
def set_bandwidth_via_cli(id_device, ip_address, username, password, ip_target, simple_que_max_down, simple_que_max_up):
    device = get_object_or_404(Device, id=id_device)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)

    try:
        ssh.connect(hostname=ip_address, username=username, password=password)

        command = f'/queue simple add target={ip_target} max-limit={simple_que_max_down}/{simple_que_max_up}'
        stdin, stdout, stderr = ssh.exec_command(command)

    except paramiko.AuthenticationException:
        raise Exception("Authentication failed.")

    except paramiko.SSHException as e:
        raise Exception(f"SSH error: {e}")

    except Exception as e:
        raise Exception(f"Error: {e}")

    finally:
        ssh.close()
```

pada fungsi code ini adalah membuat konfigurasi dengan menambahkan ip target untuk membatasi maksimal download dan uploadnya.



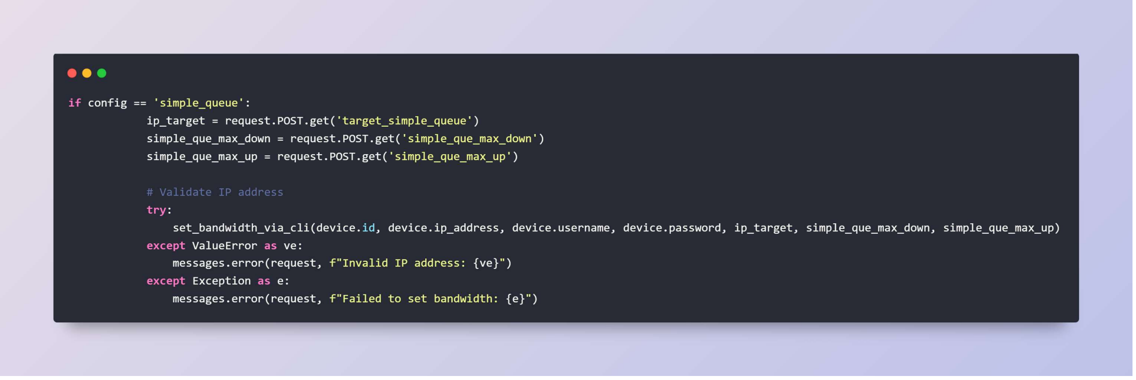
```
try:
    ssh.connect(hostname=ip_address, username=username, password=password)
```

Cara kerjanya adalah dengan mengconnectkan ssh dari mikrotiknya dengan cara menggunakan ip address mikrotik, username dan password dari mikrotik yang sudah tadi di tambahkan



```
command = f'/queue simple add target={ip_target} max-limit={simple_que_max_down}/{simple_que_max_up}'
stdin, stdout, stderr = ssh.exec_command(command)
```

Selanjutnya akan ada sebuah command untuk menambahkan sebuah ip target ke queue simple dengan max-limit download dan uplodnya sesuai dari nilai yang di input



```
if config == 'simple_queue':
            ip_target = request.POST.get('target_simple_queue')
            simple_que_max_down = request.POST.get('simple_que_max_down')
            simple_que_max_up = request.POST.get('simple_que_max_up')

            # Validate IP address
            try:
                set_bandwidth_via_cli(device.id, device.ip_address, device.username, device.password, ip_target, simple_que_max_down, simple_que_max_up)
            except ValueError as ve:
                messages.error(request, f"Invalid IP address: {ve}")
            except Exception as e:
                messages.error(request, f"Failed to set bandwidth: {e}")
```

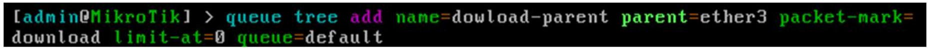
Fungsi di atas di eksekusi jika nilai dari config pada page config.html adalah sama dengan simple_queue maka ambil data dari form lalu lakukan pemanggilan fungsi di tadi dengan melempar nilai variable yang diset

C. Queue tree

Pertama melakukan sebuah configurasi manual melalui mikrotik seperti berikut

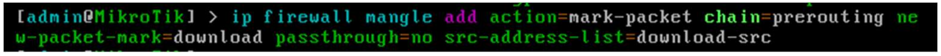
config untuk queue tree download

- Membuat parent download



```
[admin@MikroTik] > queue tree add name=dowload-parent parent=ether3 packet-mark=
download limit-at=0 queue=default
```

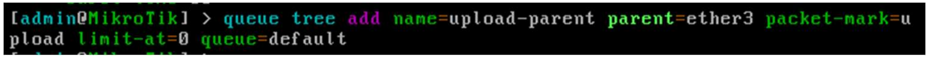
- Membuat mangle untuk menandai paket download



```
[admin@MikroTik] > ip firewall mangle add action=mark-packet chain=prerouting ne
w-packet-mark=download passthrough=no src-address-list=download-src
```

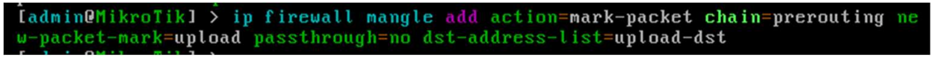
config untuk queue tree upload

- Membuat parent upload



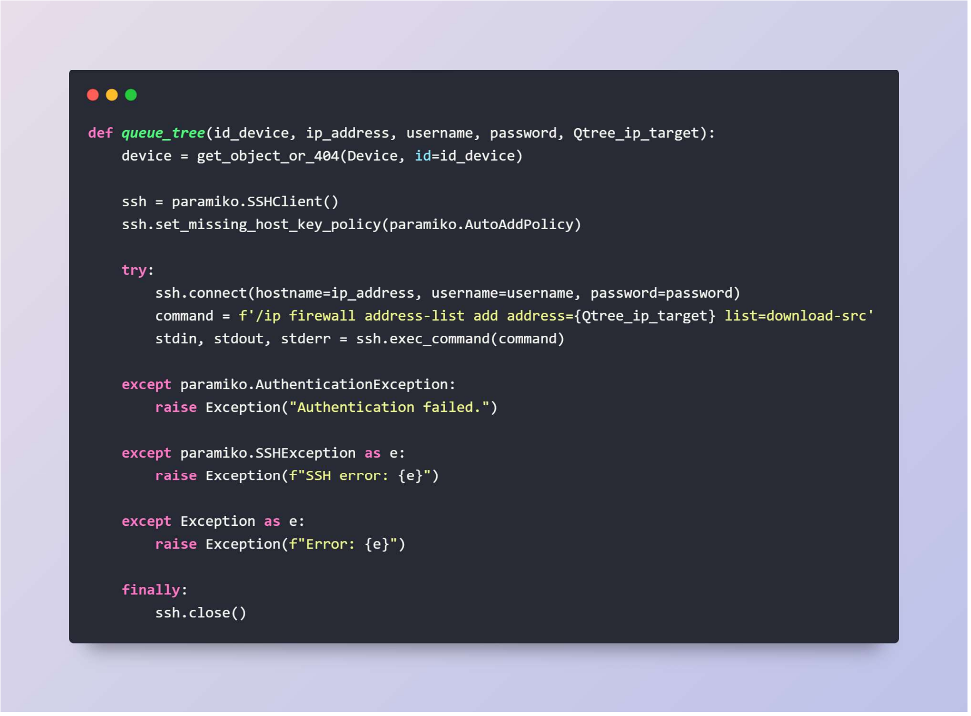
```
[admin@MikroTik] > queue tree add name=upload-parent parent=ether3 packet-mark=u
pload limit-at=0 queue=default
```

- Membuat mangle menandai paket upload



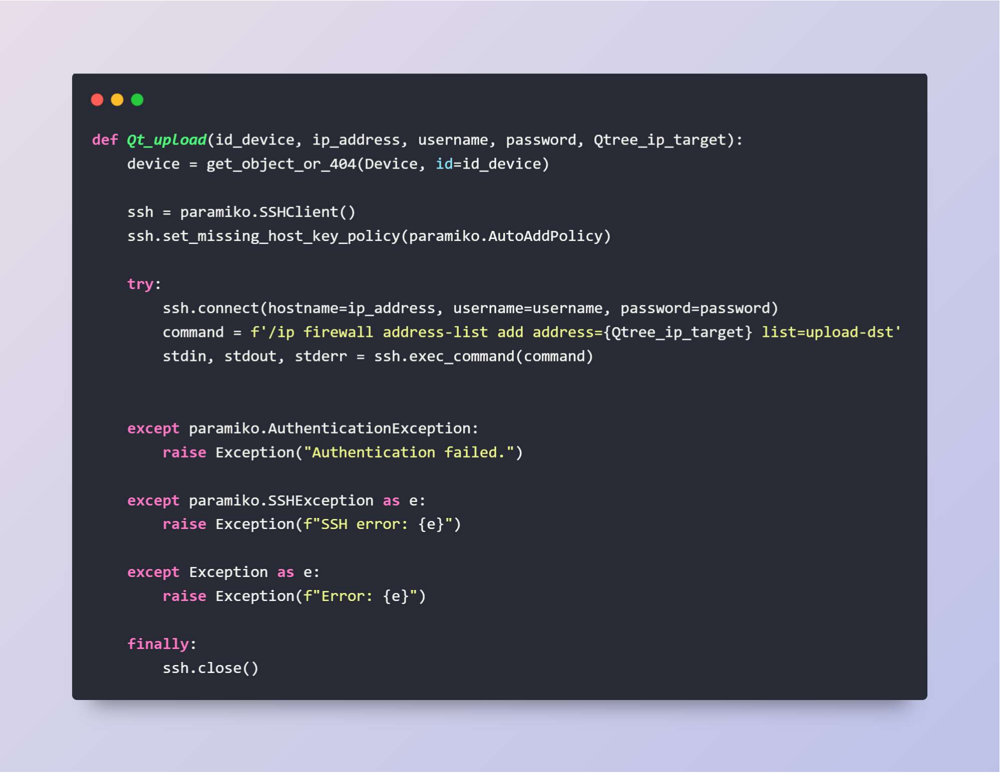
```
[admin@MikroTik] > ip firewall mangle add action=mark-packet chain=prerouting ne
w-packet-mark=upload passthrough=no dst-address-list=upload-dst
```

Setelah itu lalu membuat sebuah fungsi untuk menambhkan ip firewall address ke parent download dan upload



```python
def queue_tree(id_device, ip_address, username, password, Qtree_ip_target):
    device = get_object_or_404(Device, id=id_device)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)

    try:
        ssh.connect(hostname=ip_address, username=username, password=password)
        command = f'/ip firewall address-list add address={Qtree_ip_target} list=download-src'
        stdin, stdout, stderr = ssh.exec_command(command)

    except paramiko.AuthenticationException:
        raise Exception("Authentication failed.")

    except paramiko.SSHException as e:
        raise Exception(f"SSH error: {e}")

    except Exception as e:
        raise Exception(f"Error: {e}")

    finally:
        ssh.close()
```

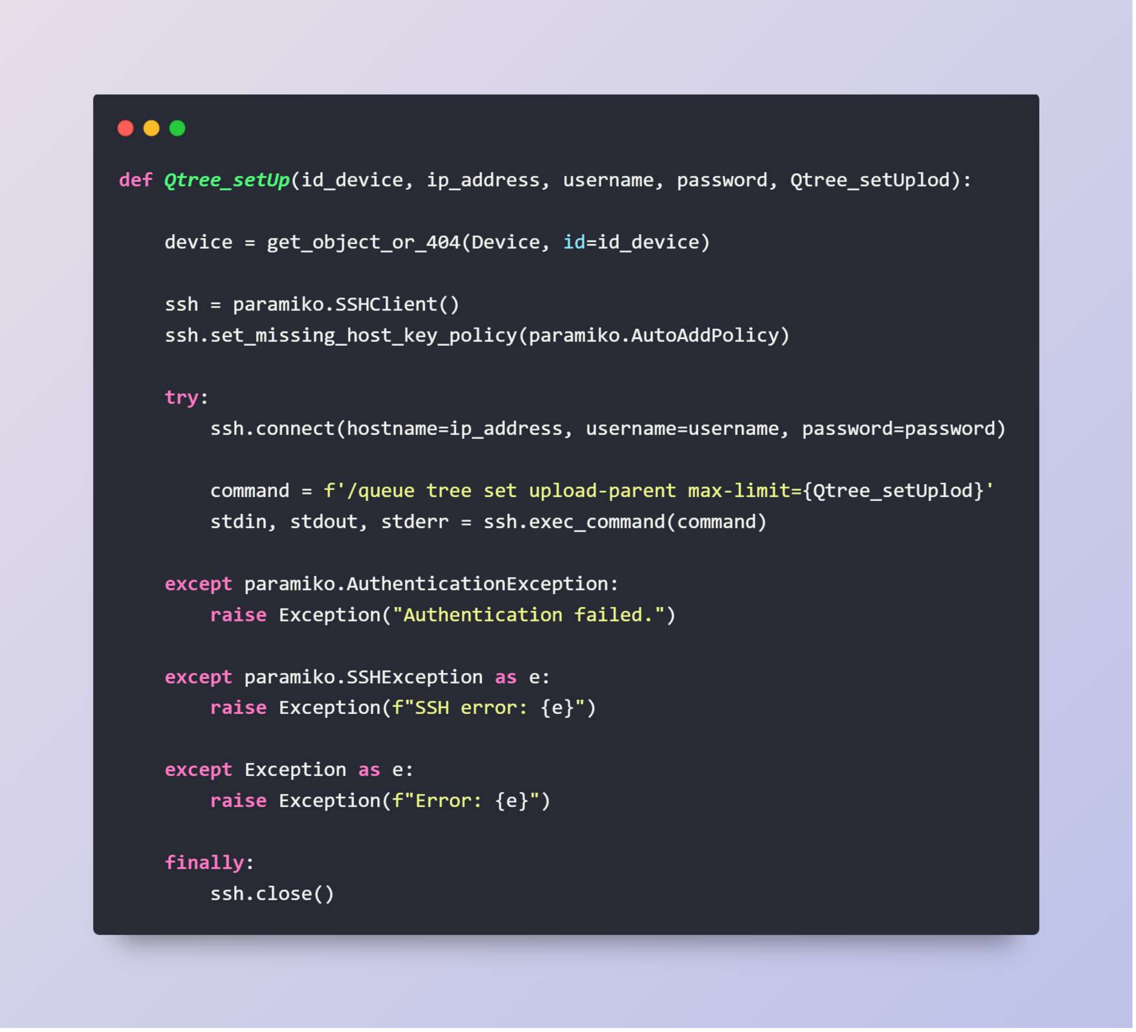
1 queue tree untuk download



```python
def Qt_upload(id_device, ip_address, username, password, Qtree_ip_target):
    device = get_object_or_404(Device, id=id_device)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)

    try:
        ssh.connect(hostname=ip_address, username=username, password=password)
        command = f'/ip firewall address-list add address={Qtree_ip_target} list=upload-dst'
        stdin, stdout, stderr = ssh.exec_command(command)


    except paramiko.AuthenticationException:
        raise Exception("Authentication failed.")

    except paramiko.SSHException as e:
        raise Exception(f"SSH error: {e}")

    except Exception as e:
        raise Exception(f"Error: {e}")

    finally:
        ssh.close()
```

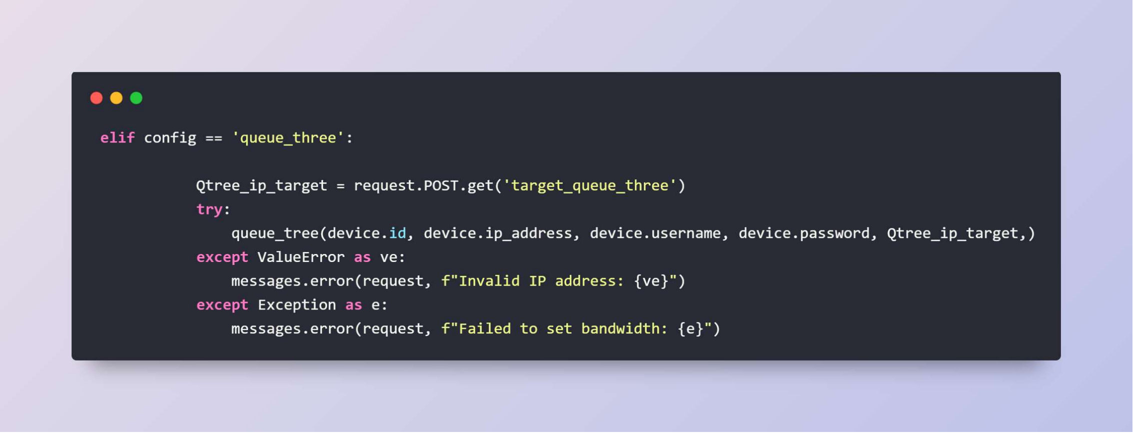
2 queue tree untuk upload

Ini merupakan sebuah fungsi yang hamper mirip dengan simple queue di atas hanya beda pada command yang hanya menambahkan ip address untuk parent download dan parent upload



```python
def Qtree_setUp(id_device, ip_address, username, password, Qtree_setUplod):

    device = get_object_or_404(Device, id=id_device)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)

    try:
        ssh.connect(hostname=ip_address, username=username, password=password)

        command = f'/queue tree set upload-parent max-limit={Qtree_setUplod}'
        stdin, stdout, stderr = ssh.exec_command(command)

    except paramiko.AuthenticationException:
        raise Exception("Authentication failed.")

    except paramiko.SSHException as e:
        raise Exception(f"SSH error: {e}")

    except Exception as e:
        raise Exception(f"Error: {e}")

    finally:
        ssh.close()
```

Ini merupakan bagian dari fungsi untuk menset limit dari max upload-parentnya. Metodenya sama mengambil nilai dari form yang di inputkan



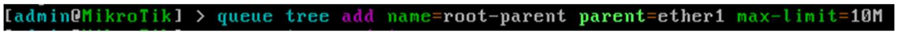
```python
elif config == 'queue_three':

            Qtree_ip_target = request.POST.get('target_queue_three')
            try:
                queue_tree(device.id, device.ip_address, device.username, device.password, Qtree_ip_target,)
            except ValueError as ve:
                messages.error(request, f"Invalid IP address: {ve}")
            except Exception as e:
                messages.error(request, f"Failed to set bandwidth: {e}")
```

Ini merupakan cara pemanggilan dari fungsi queue three Dimana fungsi itu akan di execusi jika nilai config dari device.config adalah queue_three maka jalankan fungsi queue_tree

D. HTB

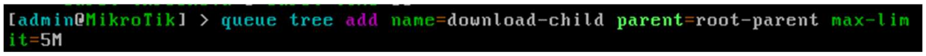
Untuk htb hamper mirip dengan queue tree configurasinya, yaitu :

- Buat main parent

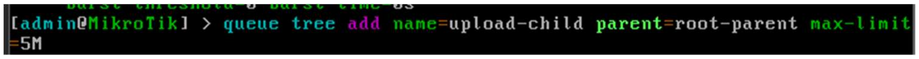


```
[admin@MikroTik] > queue tree add name=root-parent parent=ether1 max-limit=10M
```

- Mmbuat child dari download dan upload

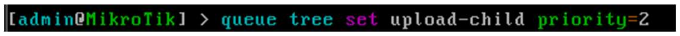


```
[admin@MikroTik] > queue tree add name=download-child parent=root-parent max-lim
it=5M
```

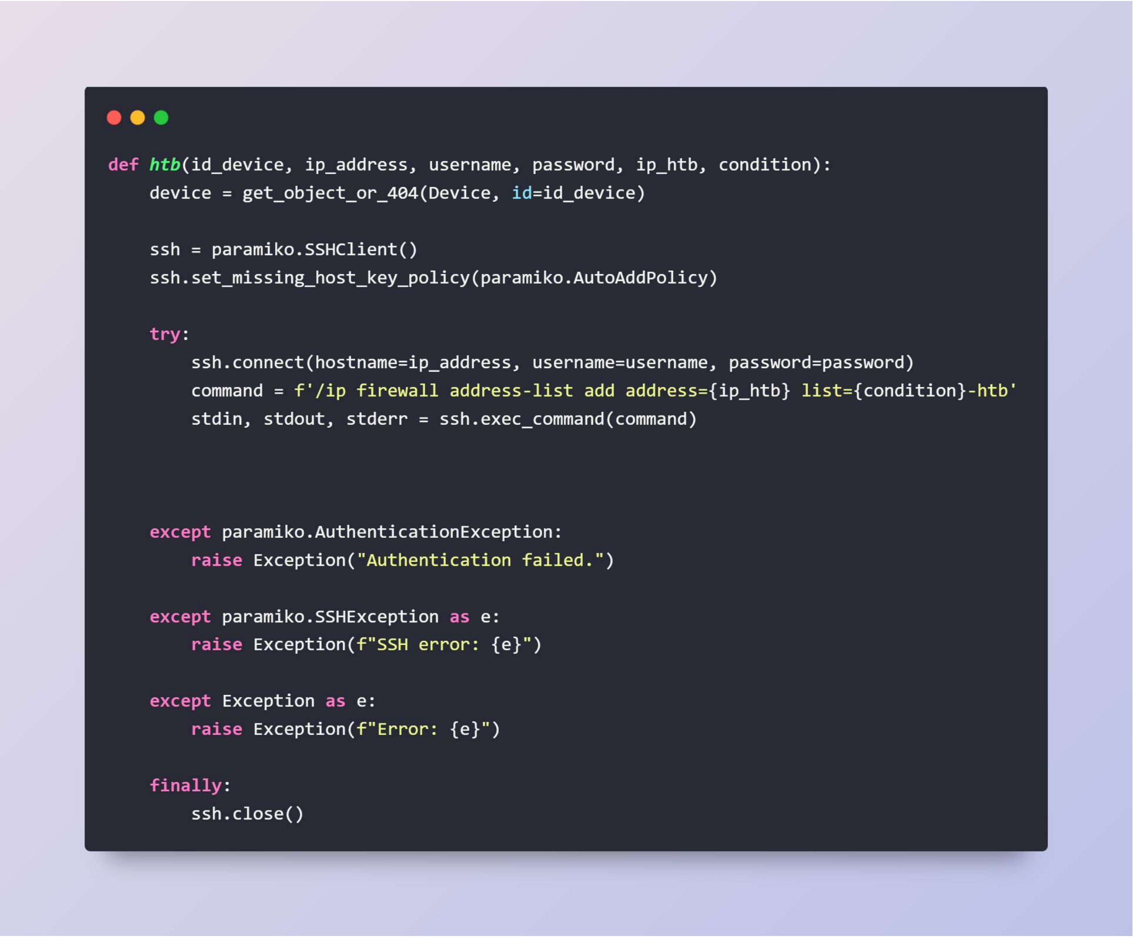


```
[admin@MikroTik] > queue tree add name=upload-child parent=root-parent max-limit
=5M
```

- Setelah itu set prioirtynya



```
[admin@MikroTik] > queue tree set upload-child priority=2
```



```python
def htb(id_device, ip_address, username, password, ip_htb, condition):
    device = get_object_or_404(Device, id=id_device)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)

    try:
        ssh.connect(hostname=ip_address, username=username, password=password)
        command = f'/ip firewall address-list add address={ip_htb} list={condition}-htb'
        stdin, stdout, stderr = ssh.exec_command(command)



    except paramiko.AuthenticationException:
        raise Exception("Authentication failed.")

    except paramiko.SSHException as e:
        raise Exception(f"SSH error: {e}")

    except Exception as e:
        raise Exception(f"Error: {e}")

    finally:
        ssh.close()
```

ini merupakan fungsi untuk menambahkan configurasi ke ip address target, agar fungsi ini dapat di eksekusi maka nilai device.config = htb, hal ini sama dengan fungsi untuk eksekusi fungsi simple queue dan queue tree